Probe-based Walk on Spheres for Efficient Path Reusing

WANCHAO HUANG*, University of Science and Technology of China, China
YUTIAN ZHU*, University of Science and Technology of China, China
QING FANG†, University of Science and Technology of China, China
LIGANG LIU, University of Science and Technology of China, China and Laoshan Laboratory, China

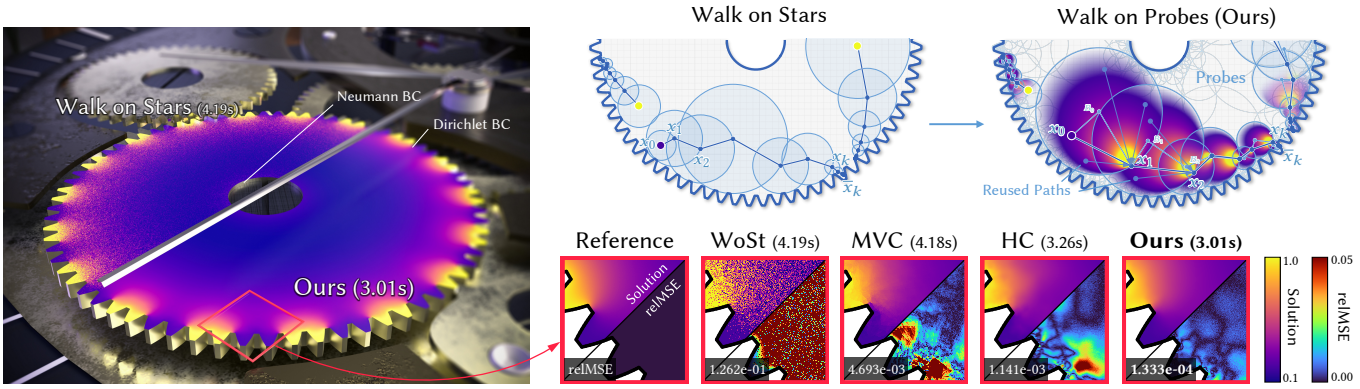


Fig. 1. We solve a 2D screened Poisson problem ($\kappa = 5.0$) with mixed Dirichlet and Neumann boundary conditions on a watch gear [The Vipron 2024]. Our Walk on Probes method achieves smooth, low-variance solutions by reusing random walk paths across spatial probes. Compared to state-of-the-art grid-free solvers like Walk on Stars (WoSt), Mean Value Caching (MVC), and Harmonic Caching (HC), our approach delivers the lowest error (relMSE: 1.333×10^{-4}) in the shortest computation time (3.01s).

The Walk on Spheres (WoS) algorithm is a mesh-free and highly flexible Monte Carlo method for solving partial differential equations, but its practical applicability is limited by slow $O(N^{-1/2})$ convergence. While prior variance reduction techniques exploit spatial correlations through integral properties of the PDE, they do not fully utilize the intrinsic Markov structure of the WoS process. We introduce a new variance reduction framework based on reusing intermediate states along each random walk. Leveraging the Markov property, we show that every point visited by a WoS trajectory provides a valid unbiased estimator, but its direct use is hindered by the complex distribution induced by dynamically generated spheres. To resolve this, we propose the Walk on Probes (WoP) algorithm, which replaces dynamic spheres with a set of fixed, pre-distributed spherical probes inside the domain. This converts the intractable distribution of path points into samples on fixed boundaries, enabling efficient evaluation through the Poisson integral formula. We further develop a specialized method that combines control variates with self-normalization to further reduce variance. Together, these components substantially improve sample efficiency while preserving the flexibility of WoS. Code and data for this paper are at https://github.com/USTCGCL-WoS/Walk-on-Probes.

*Both authors contributed equally to this research.
†Corresponding author.

Authors' Contact Information: Wanchao Huang, hwc20040629@mail.ustc.edu.cn, University of Science and Technology of China, China; Yutian Zhu, zhuyutian@mail.ustc.edu.cn, University of Science and Technology of China, China; Qing Fang, fq1208@mail.ustc.edu.cn, University of Science and Technology of China, China; Ligang Liu, lgliu@ustc.edu.cn, University of Science and Technology of China, China and Laoshan Laboratory, China.

ACM 1557-7368/2026/7-ART129
https://doi.org/10.1145/3811366

CCS Concepts: • **Mathematics of computing → Partial differential equations**.

ACM Reference Format:
Wanchao Huang, Yutian Zhu, Qing Fang, and Ligang Liu. 2026. Probe-based Walk on Spheres for Efficient Path Reusing. *ACM Trans. Graph.* 45, 4, Article 129 (July 2026), 12 pages. https://doi.org/10.1145/3811366

1 Introduction

In recent years, the Walk on Spheres (WoS) algorithm [Muller 1956; Sawhney and Crane 2020] has emerged as a prominent Monte Carlo method for solving Partial Differential Equations (PDEs). Unlike traditional grid-based methods, WoS possesses several compelling advantages: it is entirely mesh-free, inherently parallelizable, and supports local evaluation of the solution without solving for the entire domain. These characteristics allow WoS to avoid the discretization artifacts and meshing complexities associated with Finite Element Methods (FEM) or Finite Difference Methods (FDM), demonstrating superior robustness and flexibility in handling complex geometries.

As is well known, WoS faces a challenge inherent to Monte Carlo algorithms: slow convergence. Its estimation error decreases at a rate of $O\left(N^{-1/2}\right)$, where N is the number of random samples. This slow convergence limits the utility of WoS in scenarios requiring high-precision solutions. To mitigate this issue, previous works have explored various variance reduction strategies. Sawhney and Crane [2020] employed control variates and importance sampling to reduce the variance of single-point estimates. Subsequent algorithms leveraged spatial correlations via caching: while Boundary Value Caching [Miller et al. 2023] and Mean Value Caching [Bakbouk

and Peers 2023] store information at discrete sample points, Harmonic Caching [Zhou et al. 2025] introduces a structure similar to global illumination probes, using a set of fixed covering spheres to reconstruct the solution field. However, these strategies treat the underlying WoS process merely as a black-box solver to evaluate boundary samples, ignoring the rich information contained within the random walk trajectories.

In this paper, we propose a novel variance reduction technique that revisits the random walk process. Our core insight stems from the *Markov property*: a single WoS trajectory provides an unbiased estimate not only for the starting point but for every intermediate point along the path. Theoretically, reusing these path points could improve sample efficiency. However, in standard WoS, the spheres are generated dynamically based on the distance to the boundary. This lack of a persistent structure prevents the efficient aggregation of intermediate estimates.

To overcome this, we propose the *Walk on Probes* (WoP) algorithm. Similar to Harmonic Caching, we pre-distribute a set of fixed spheres—referred to as probes—within the domain to cache intermediate results. Then we perform off-center random walks transitioning among the boundaries of these probes rather than generating dynamic spheres. Unlike the disposable spheres in WoS, fixed probes accumulate samples over time, allowing for efficient solution reconstruction via series expansions. Furthermore, we introduce a specialized variance reduction method that combines control variates with self-normalization to further stabilize the estimates.

Our main contributions are as follows:

- The *Walk on Probes* (WoP) framework, which enables a novel path reuse strategy by performing random walks on probes, transforming intermediate path steps into valid, reusable boundary samples.
- A specialized estimator incorporating *self-normalization* and *control variates* to minimize variance.

2 Related Work

Grid-Free Monte Carlo PDE Solvers. The connection between PDEs and stochastic processes, established by Kakutani [1944a], was made computationally tractable by Muller [1956] via the Walk on Spheres (WoS) algorithm. WoS enables efficient, grid-free sampling by taking maximal steps to the boundary of the largest empty sphere. Recently, Sawhney and Crane [2020] demonstrated the robustness of WoS for complex, mesh-free geometries in graphics, sparking numerous extensions. These include adaptations for spatially varying coefficients [Sawhney et al. 2022], Neumann and Robin boundary conditions via Walk on Stars [Miller et al. 2024b; Sawhney et al. 2023], and differentiable formulations for inverse problems [Miller et al. 2024a; Yilmazer et al. 2024; Yu et al. 2024]. In addition to these variants, the Walk on Fixed Spheres method replaced transient spheres with fixed spheres [Sabelfeld et al. 2008, 2006, 2004]. Similarly, Himmler and Günther [2025] used fixed semicircles and sectors rather than spheres. Alternatively, Sugimoto et al. [2023] introduced Walk on Boundary (WoB) to solve boundary integral equations via recursive random walks on boundary.

Variance Reduction. As Monte Carlo estimators, WoS methods suffer from slow $O(N^{-1/2})$ convergence, making variance reduction essential. Mean Value Caching (MVC) [Bakbouk and Peers 2023] and Boundary Value Caching (BVC) [Miller et al. 2023] store intermediate results to estimate solutions, though they face challenges regarding boundary correlation artifacts and numerical instability, respectively. Recent approaches leverage off-center Poisson Kernel Splatting to reuse walk information. Qi et al. [2022] proposed a bidirectional formulation (BDWoS), while others focused on reusing the first step of random walks [Bao et al. 2025; Czekanski et al. 2024]. Most recently, Harmonic Caching [Zhou et al. 2025] improved efficiency by employing a Fourier formulation of Poisson Kernel Splatting.

Probe-based Rendering Techniques. In computer graphics, caching radiometric data to accelerate Global Illumination (GI) is a foundational strategy. Pioneered by Irradiance Caching [Ward et al. 1988] and Radiance Caching [Krivanek et al. 2005], these methods reduce variance via spatial interpolation. To further decouple lighting from geometry, Greger et al. [1998] introduced Irradiance Volumes, laying the groundwork for probe-based rendering. This evolved through Precomputed Radiance Transfer (PRT) [Sloan et al. 2002], which projected signals onto basis functions, and modern real-time techniques like Dynamic Diffuse Global Illumination (DDGI) [Majercik et al. 2019]. These probes act as geometric proxies that filter high-variance Monte Carlo samples through temporal accumulation and spatial interpolation. While designed for lighting, this philosophy of constraining stochastic samples to analytically tractable geometries directly inspires our use of probes for solving PDEs.

Our Approach. Distinct from the aforementioned methods, we revisit the intrinsic Markov Property of the random walk process as a fundamental source of variance reduction. While recent works have attempted to reuse the first step of the walk [Bao et al. 2025; Czekanski et al. 2024], effectively leveraging the entire trajectory remains an open challenge due to the intractable joint distribution of dynamically generated spheres in standard WoS. In this paper, we address this limitation by introducing the Walk on Probes (WoP) framework. Similar to the Walk on Fixed Spheres method [Sabelfeld et al. 2008, 2006, 2004], instead of relying on random maximal spheres, we constrain the random walk to a set of pre-distributed spherical probes. This geometric regularization transforms the stochastic path points into valid boundary samples, enabling efficient information reuse via the Poisson integral formula.

3 Method

3.1 Problem Statement

We consider the mixed boundary value problem for the screened Poisson equation in a bounded domain $\Omega \subset \mathbb{R}^d$. The boundary $\partial\Omega$ is partitioned into disjoint Dirichlet ($\partial\Omega_{\mathrm{D}}$) and Neumann ($\partial\Omega_{\mathrm{N}}$) segments. The problem is formulated as follows:

$$\begin{aligned} \Delta u(x) - \kappa^2 u(x) &= f(x) \quad \text{in } \Omega, \\ u(x) &= g(x) \quad \text{on } \partial\Omega_{\mathrm{D}}, \\ \frac{\partial u(x)}{\partial \mathbf{n}} &= h(x) \quad \text{on } \partial\Omega_{\mathrm{N}}, \end{aligned} \tag{1}$$

where n denotes the outward unit normal vector, $\kappa^2 \geq 0$ is a constant. For clarity, we primarily focus on the Laplace ($\kappa^2 = 0, f = 0$)

equation in two-dimensional space. We discuss the additional procedure for handling Poisson ($\kappa^2 = 0$) and screened Poisson ($\kappa^2 > 0$) equations in Section 3.3.5 and Section 3.3.6.

3.2 Path Reuse in Classical WoS

3.2.1 Classical Walk on Spheres (WoS). The Walk on Spheres (WoS) algorithm is based on the Poisson integral formula [Evans 2022] within a ball. For any ball $B_c = B(c, R_c) \subset \Omega$ centered at c with radius R_c, the solution at the center can be expressed using the Poisson integral formula:

$$u(c) = \int_{\partial B_c} u(z) P^{B_c}(c, z)\, \mathrm{d}S(z), \tag{2}$$

where $P^{B_c}(c, z)$ is the Poisson kernel. In the classical WoS, R_c is typically chosen as the distance to the boundary, i.e., $R_c = d(c, \partial\Omega)$. This integral yields a recursive Monte Carlo estimator:

$$\hat{u}(c; B_c) = \frac{1}{N} \sum_{i=1}^{N} \frac{P^{B_c}(c, Z_i)}{p_i(Z_i)} \hat{u}(Z_i). \tag{3}$$

where Z_i is sampled from a probability density $p_i(z)$ on ∂B_c. In the standard implementation, Z is sampled uniformly, i.e., $p(z) = 1/|\partial B_c|$. Since $P^{B_c}(c, z)$ is also constant $(1/|\partial B_c|)$ for the Laplace equation at the ball center, the terms cancel out, simplifying to $\hat{u}(c) = \frac{1}{N} \sum_{i=1}^{N} \hat{u}(Z_i)$.

The recursion continues until the walker reaches the ϵ-shell of the Dirichlet boundary, i.e., $d(c, \partial\Omega_{\mathrm{D}}) < \epsilon$, at which point the walk terminates with $\hat{u}(c) \approx g(\bar{c})$, where $\bar{c}$ is the closest point on $\partial\Omega$ to c. Fig. 3a shows the process.

3.2.2 Off-Center Estimator. Standard WoS strictly evaluates the solution at the center of the ball. However, the Poisson integral formula is valid for *any* point x located inside the ball B_c, not merely the center. For an arbitrary query point $x \in B_c$, the solution is given by the off-center Poisson integral formula:

$$u(x) = \int_{\partial B_c} u(z) P^{B_c}(x, z)\, \mathrm{d}S(z). \tag{4}$$

This allows us to construct an off-center estimator:

$$\hat{u}(x; B_c) = \frac{1}{N} \sum_{i=1}^{N} \frac{P^{B_c}(x, Z_i)}{p_i(Z_i)} \hat{u}(Z_i). \tag{5}$$

Unlike the centered case, the Poisson kernel $P^{B_c}(x, z)$ here depends on the relative position of x and is generally not constant. As a result, a uniform $p(z)$ will cause high variance.

3.2.3 Generalized Weighted Estimator. Building on the off-center observation, we define the *neighbor set* $\mathcal{N}_x$ for a query point x as the set of all valid centers whose associated maximum inscribed balls cover x:

$$\mathcal{N}_x = \{c \in \Omega \mid x \in B(c, d(c, \partial\Omega))\}. \tag{6}$$

Since any ball centered at $c \in \mathcal{N}_x$ provides a valid unbiased estimator for $u(x)$, we can construct a generalized estimator by averaging over these overlapping domains. Let $\alpha_x(c)$ be a normalized weight function satisfying $\int_{\mathcal{N}_x} \alpha_x(c)\, \mathrm{d}c = 1$. The solution $u(x)$ can be expressed as a weighted superposition:

$$u(x) = \int_{\mathcal{N}_x} \alpha_x(c) \left[\int_{\partial B_c} u(z) P^{B_c}(x, z)\, \mathrm{d}S(z) \right] \mathrm{d}c. \tag{7}$$

In practice, assume we aggregate N balls with centers $\{C_i\}_{i=1}^{N}$, and for each ball we sample a boundary point $Z_i \sim p(z \mid C_i)$, the generalized estimator becomes:

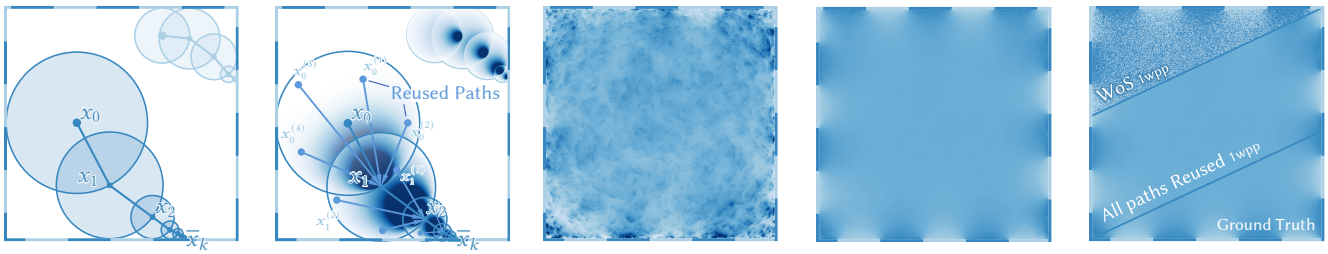
$$\hat{u}_{\mathrm{gen}}(x) = \frac{\sum_{i=1}^{N} w_x(C_i) \hat{u}(x; B_{C_i})}{\sum_{i=1}^{N} w_x(C_i)}, \tag{8}$$

where $w_x(C_i)$ represents the discretized weight corresponding to $\alpha_x(c)$.

3.2.4 Naive Path Reuse. With the formulation of the generalized weighted off-center estimator, it is straightforward to extend the reuse of neighboring WoS processes. Specifically, for a given query point x, we can treat neighboring query points as potential centers $\{C_i\}$ and reuse their respective random walks to estimate $u(x)$, effectively reducing variance. This strategy has been successfully demonstrated in previous works [Bao et al. 2025; Czekanski et al. 2024]. However, these approaches do not fully exploit the available information, as they restrict reuse to the initial step of the walk, overlooking the information contained in intermediate steps along the path.

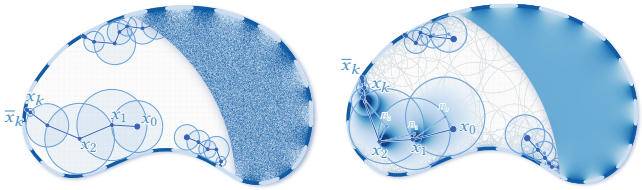
Since the random walks generated by the WoS algorithm satisfy the Markov property, the future trajectory depends solely on the current position. Consequently, every intermediate point along a path can serve as a valid starting point for an unbiased estimator. While characterizing the spatial distribution of path points analytically is challenging, we can explicitly record their generation process. During the random walk, we cache the key components for each step: the ball used for the jump, the coordinates of the sampled boundary point, and the corresponding probability density function (PDF) value. Upon termination, the boundary value is propagated back to compute estimates for all recorded points. This process yields a sequence of *path balls*, which encapsulate the complete information of the trajectory. By applying Eq. (5), we can provide an unbiased estimate for any query point located within these path balls. Furthermore, by aggregating a larger collection of path balls into the generalized estimator (Eq. (8)), we can significantly reduce the variance. Fig. 2 show the process.

3.2.5 Complexity Analysis. While reusing full path trajectories maximizes information utilization, it introduces a significant computational bottleneck. To ensure a fair comparison, consider a scenario where one walk is initiated from each of the Q query points, resulting in a total of $O(Q)$ path balls. In the traditional WoS framework, the complexity scales linearly, $O(Q)$, as each query point only processes its own trajectory. In contrast, the generalized weighted estimator requires aggregating contributions from all covering path balls for each query point. Since the total number of path balls is proportional to Q, and each query point may be covered by a number of balls that also scales with Q, evaluating the Poisson kernel for all such overlaps leads to an overall computational complexity of $O(Q^2)$.



(a) WoS Paths (b) Reused (c) 1024 paths reused (d) All paths reused (e) Comparison with WoS

Fig. 2. Procedure of naive path-reused WoS. (a) A standard WoS path. (b) Illustration of the path reuse mechanism: a single walk path can provide estimates for any query point contained within its sequence of path balls. Points contained by multiple path balls accumulate multiple estimates. (c-d) More reused paths yield a smoother result. (e) Comparison between standard WoS and path-reused WoS using 1 walk per pixel (wpp). Path reuse significantly reduces variance while maintaining the same number of total walks.



(a) Walk on Spheres (b) Walk on Probes

Fig. 3. (a) Walk on Spheres (WoS) constructs a centered sphere dynamically at each step, while (b) Walk on Probes (WoP) retrieves a covering probe. Similarly, WoP can also enable path reuse, where a single probe-based path contributes estimates to all points contained within its sequence of probes.

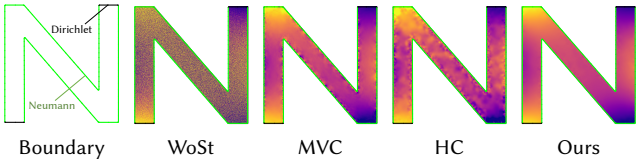


Fig. 4. Results on a narrow, Neumann-dominated scene (~140s). Here, extensive reflections cause high variance in WoSt. The narrow geometry limits sample reuse in MVC and HC, whereas our method extracts more samples from the long paths, leading to a much more stable solution.

3.3 Walk on Probes

3.3.1 Walking Method. To mitigate the computational overhead of the aforementioned strategy, we propose the *Walk on Probes (WoP)* method. The core intuition is to transition from generating transient, dynamic spheres to utilizing a pre-computed set of fixed spherical probes $\mathcal{P}$ that cover the region of interest (ROI).

Traditionally, the WoS algorithm requires spheres to be centered exactly at the current walker position. However, leveraging the generalized off-center estimator (Eq. (5)), we can formulate a walking strategy on probes:

(1) **Termination:** If x is within the ϵ-shell of the Dirichlet boundary, terminate the walk.
(2) **Probe Selection:** Identify a probe B_c covering x. If no probe is available, fall back to the standard WoS or Walk on Stars (WoSt) step (centering a sphere or star-shaped region at x).
(3) **Sampling:** Sample the next position Z on the spherical boundary ∂B_c and update $x \leftarrow Z$.

The process is shown in Fig. 3b. The primary difference between WoP and standard WoS is that the Poisson kernel is no longer uniform over the boundary, which would lead to high variance if uniform sampling were used. To address this, we employ importance sampling based on the Poisson kernel to reduce variance.

This walking method is unbiased, since it is based on the unbiased off-center estimator Eq. (5), save for the negligible bias introduced by the ϵ-shell, similar to standard WoS.

An alternative understanding of the unbiasedness of this walking method is provided by the Feynman-Kac formula [Kac 1949] (also known as Kakutani's Principle [Kakutani 1944b] for the Laplace equation). In this view, the solution $u(x)$ for any $x \in \Omega$ is formulated as $\mathbb{E}_x[g(y)]$, where $y \in \partial\Omega$ is the first hitting point of a Wiener process starting at x. In Eq. (4), the Poisson kernel $P^{B_c}(x, z)$ represents the exact PDF of the first hitting point on the probe boundary ∂B_c. By recursively walking on probe boundaries, we simulate the Wiener process reaching the domain boundary, which is identical to standard WoS. Therefore, recursively evaluating Eq. (5) yields a strictly unbiased estimate of this process.

By adopting this formulation, we reduce the number of path balls from a potentially infinite number to a finite set. Instead of generating transient balls for every step of every walk, we map all random walk steps to a finite set of probes. Since the inherent Markov property is retained, the path points still provide unbiased estimates. This allows us to aggregate a large number of intermediate path points as boundary samples on each probe, whereas naive path reuse has only one sample per path ball. After aggregating all the path points, we can leverage Eq. (5) to evaluate any point inside the probe. This strategy significantly improves convergence (e.g., Fig. 4).

In order to efficiently utilize these boundary samples, we analyze two caching strategies to manage the information on these probes: *Poisson Kernel Caching* and *Harmonic Caching* [Zhou et al. 2025].

3.3.2 Poisson Kernel Caching. We refer to the approach of explicitly storing all boundary samples and evaluating interior query points via Eq. (5) as Poisson Kernel Caching. This method directly uses the Poisson kernel to propagate boundary values to the interior domain. Assuming there are Q query points and Q corresponding walks, generating $O(Q)$ path points for each probe, the computational complexity for evaluation scales to $O(Q^2)$, which becomes prohibitive for large-scale problems.

3.3.3 Harmonic Caching. As described by Zhou et al. [2025], the solution of a Laplace equation in a probe can be reconstructed using a Fourier series representation:

$$u(r,\theta;B_c) = a_0^{B_c}\mathcal{R}^0(r,R_c) + 2\sum_{l=1}^{\infty}\mathcal{R}^l(r,R_c)\left(a_l^{B_c}\cos(l\theta) + b_l^{B_c}\sin(l\theta)\right), \tag{9}$$

where $a_l^{B_c}$ and $b_l^{B_c}$ are Fourier coefficients and $\mathcal{R}^l(r,R_c) = (r/R_c)^l$ is the radial basis function. Although truncating the Fourier series at a finite order introduces bias, the rapid convergence of the series ensures that the error contribution from these high-order terms is negligible.

The Fourier coefficients can be estimated by Monte Carlo Integration:

$$\hat{a}_l^{B_c} = \frac{1}{N}\sum_{i=1}^{N}\frac{\hat{u}(R_c,\Theta_i)\cos(l\Theta_i)}{2\pi p_i(\Theta_i)}, \quad \hat{b}_l^{B_c} = \frac{1}{N}\sum_{i=1}^{N}\frac{\hat{u}(R_c,\Theta_i)\sin(l\Theta_i)}{2\pi p_i(\Theta_i)}, \tag{10}$$

where N is the number of samples and $p_i(\theta)$ denotes the sampling PDF with respect to the angle, corresponding to the spatial PDF $p_i(z)$ on the boundary. In 3D, the expansion employs spherical harmonics [Zhou et al. 2025, Appendix A].

This method decouples the number of boundary samples from the evaluation cost. The process involves two stages:

(1) **Projection:** We first aggregate the boundary path points into Fourier coefficients of order L. For $O(Q)$ path points, this incurs a cost of $O(QL)$.
(2) **Evaluation:** We then compute the solution at each of the Q query points using these pre-computed coefficients, with a cost of $O(QL)$.

The total complexity is drastically reduced to $O(Q)$. Harmonic Caching transforms the expensive quadratic complexity into a linear one, making high-quality global reuse feasible.

3.3.4 Variance Reduction: Control Variates and Self-normalization. To mitigate variance in the Fourier formulation, we employ a hybrid strategy combining *self-normalized importance sampling (SNIS)* and *control variates*. Let us define the importance weight for the i-th sample as $w_i = [2\pi p_i(\Theta_i)]^{-1}$. We normalize the estimator using the sum of weights $W_\Sigma = \sum_{i=1}^{N} w_i$.

For the zero-order coefficient ($l = 0$), we apply SNIS directly to estimate the DC component $\hat{a}_0^{B_c}$:

$$\hat{a}_0^{B_c} = \frac{1}{W_\Sigma}\sum_{i=1}^{N} w_i\hat{u}(R_c,\Theta_i). \tag{11}$$

For higher-order coefficients ($l \geq 1$), we utilize $\hat{a}_0^{B_c}$ as a *control variate*. Since the integral of a constant against sinusoidal basis functions vanishes (i.e., $\int \hat{a}_0^{B_c}\cos(l\theta)d\theta = 0$), subtracting this DC component does not alter the expectation but significantly reduces the variance by minimizing the integrand's magnitude (or residual energy). The combined estimators are:

$$\hat{a}_l^{B_c} = \frac{1}{W_\Sigma}\sum_{i=1}^{N} w_i(\hat{u}(R_c,\Theta_i) - \hat{a}_0^{B_c})\cos(l\Theta_i), \quad \hat{b}_l^{B_c} = \frac{1}{W_\Sigma}\sum_{i=1}^{N} w_i(\hat{u}(R_c,\Theta_i) - \hat{a}_0^{B_c})\sin(l\Theta_i). \tag{12}$$

While SNIS introduces a negligible bias for finite N, the estimator remains asymptotically consistent. As shown in Fig. 8, this combined approach significantly smooths the solution and reduces error when the sample distribution deviates from the ideal target distribution.

3.3.5 Poisson Equations. To extend this method to Poisson equations, we need to evaluate the contribution of the source term. In traditional WoS, we sample M source samples $\{Y_j \in B_c\}_{j=1}^{M}$ every step to compute the source contribution:

$$\hat{u}(c) = \frac{1}{N}\sum_{i=1}^{N}\frac{P^{B_c}(c,Z_i)}{p_i(Z_i)}\hat{u}(Z_i) + \frac{1}{M}\sum_{j=1}^{M}\frac{G^{B_c}(c,Y_j)}{q_j(Y_j)}f(Y_j), \tag{13}$$

where $q_j(y)$ is the PDF value. We can use the same method to evaluate it in the off-center formula:

$$\hat{u}(x;B_c) = \frac{1}{N}\sum_{i=1}^{N}\frac{P^{B_c}(x,Z_i)}{p_i(Z_i)}\hat{u}(Z_i) + \frac{1}{M}\sum_{j=1}^{M}\frac{G^{B_c}(x,Y_j)}{q_j(Y_j)}f(Y_j). \tag{14}$$

In WoP, we use a single-sample estimator during the walking process. Once the path points are collected, we sample points within each probe—with a count proportional to its volume—to estimate the values at the query points.

3.3.6 Screened Poisson Equations. The WoP estimator for screened Poisson equations follows the same structure as that for Poisson equations, with the modification that the Green's function is replaced by the *Yukawa potential* [Elepov and Mikhailov 1969] and the Poisson kernel is adjusted accordingly.

Another difference is that the radial function in Eq. (9) is dependent on κ. If $\kappa^2 > 0$, $\mathcal{R}^l(r,R_c)$ will be replaced with:

$$\mathcal{R}_\kappa^l(r,R_c) = \frac{I_l(\kappa r)}{I_l(\kappa R_c)}, \tag{15}$$

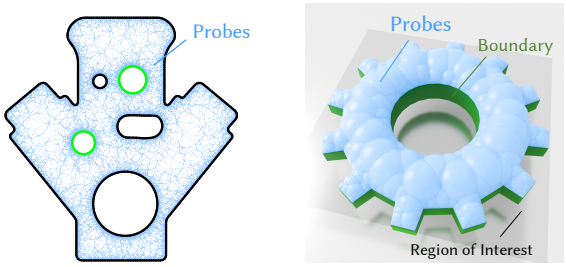
where I_l denotes the modified Bessel function of the 1st kind. As discussed in [Zhou et al. 2025, Appendix B], the radial function needs another modification in 3D.

4 Implementation

In this section, we detail the implementation of our proposed method, encompassing the generation of query points and probes, the random walk procedure, and the final solution reconstruction. We also discuss data structures for spatial acceleration and parallelization strategies to ensure computational efficiency.

4.1 Query Points Generation

Query points are defined as the discrete locations within the Region of Interest (ROI) where the solution is evaluated. For simplicity and uniformity, we generate a Cartesian grid over the entire bounding box of the domain. We then retain those points that strictly lie within the ROI to serve as our set of query points.



(a) 2D Probes (b) 3D Probes

Fig. 5. Probe placement results. The distribution is optimized to balance reconstruction accuracy and walking efficiency by maintaining sufficient overlap among probes. (a) 2D circular probes. (b) 3D spherical probes.

4.2 Placement of Probes

The placement of probes is critical for two reasons: (1) ensuring every query point is sufficiently covered for accurate solution reconstruction; and (2) preventing non-uniform distribution of boundary samples by minimizing the distance between a walk's starting position and the center of its corresponding probe.

We adopt a greedy coverage strategy based on [Zhou et al. 2025] with modifications to accommodate variance reduction requirements. Similarly we define a smooth covering weight function $w(x; B_c)$ for a point x inside a probe B_c with center c and radius R_c:

$$w(x; B_c) = 3\left(1 - \frac{\|x-c\|}{R_c}\right)^2 - 2\left(1 - \frac{\|x-c\|}{R_c}\right)^3, \tag{16}$$

where $\|\cdot\|$ denotes the L_2 norm (Euclidean distance). This polynomial ensures a smooth transition of contributions from overlapping probes. To balance reconstruction accuracy and walking efficiency, we introduce two user-defined parameters: the *reconstruction ratio* α_{rec} and the *walking ratio* α_{walk}.

- α_{rec} restricts the effective reconstruction region to the inner part of probes (e.g., ignoring the boundary layer where truncation error is higher). We empirically set $\alpha_{\text{rec}} = 0.9$.
- α_{walk} restricts the "optimal" walking region to minimize the eccentricity of the starting point, as off-center walks yield higher variance.

The generation algorithm proceeds by shuffling all query points and iterating through them. For each point, we check its cumulative weight provided by existing probes using an effective coverage radius $R_{\text{eff}} = \min(\alpha_{\text{rec}}, \alpha_{\text{walk}}) \cdot R_c$. If the cumulative weight falls below a prescribed threshold $w_{\min}$ (set to 1.0 empirically), a new probe is centered at this point, and the weights of all affected neighbors are updated. Fig. 5 shows an example.

4.3 Adaptive Spatial Indexing

To efficiently identify the most suitable probe for the next random walk step, we construct an adaptive spatial data structure (e.g., a Quadtree or Octree). Each leaf node maintains a list of candidate probes that have the potential to be the optimal probe. During the walk, for a query point x, we search for an *optimal probe* B_{c^*}. A probe B_c is considered a valid candidate only if the query point falls within its effective walking region, i.e., $|x-c|/R_c < \alpha_{\text{walk}}$. Among all valid candidates, the optimal probe is defined as the one minimizing the eccentricity:

$$c^* = \underset{c:\, |x-c|/R_c < \alpha_{\text{walk}}}{\arg\min} \left(\frac{|x-c|}{R_c}\right). \tag{17}$$

If no probe satisfies the condition, the query returns null, signaling the algorithm to switch to the fallback strategy.

4.4 Walking Procedure

The random walk process involves adaptive transitions that alternate between using pre-computed probes and on-the-fly Walk on Stars (WoSt) steps. The WoSt fallback mechanism enables our method to handle Neumann boundaries, at the cost of discarding the path sample of fallback steps.

Initialization: For each probe B_c, we generate a set of walking paths starting from its boundary ∂B_c. The number of paths, N_s, is proportional to the probe's surface area: $N_s = \max(\lambda R_c^{d-1}, N_{\min})$, where d is the dimension and λ is a density factor. Empirically, we set $N_{\min} = 12$ and select λ based on the scene scale. These walks serve to evaluate coefficients not only for B_c itself but also for other probes encountered along the trajectory.

Step Transition Strategy: Given a current position x, the next step is determined by the availability of an optimal probe:

(1) **Probe Step:** If a valid probe B_{c^*} is available, we utilize it with probability $1 - p_{\text{fb}}$.
(2) **Fallback WoSt Step:** Otherwise, we revert to the standard WoSt method. This covers cases where no probe satisfies the criteria, the Neumann boundary is too close ($d(c, \partial\Omega_{\text{N}}) < \epsilon$), or the probabilistic check (probability p_{fb}) fails.

Sampling Details within Probes: given a current position x inside a probe B_c:

(1) **Boundary Sampling:** The next position $z \in \partial B_c$ is sampled via the inverse transform method according to the normalized Poisson kernel $P^{B_c}(x, z)$.
(2) **Source Term Estimation:** An interior point $y \in B_c$ is sampled using the normalized Green's function $G^{B_c}(x, y)$ as the PDF. We distinguish two cases: **(i)** If $x = c$, we sample y analytically due to radial symmetry. **(ii)** If $x \neq c$, we employ rejection sampling: a candidate y is drawn from the Green's function of an auxiliary sphere B_x^a centered at x that encloses B_c ($B_c \subseteq B_x^a$). The candidate is accepted with probability $G^{B_c}(x, y)/G^{B_x^a}(x, y)$.

Sampling z according to the Poisson kernel is essential for maintaining estimator efficiency. Deviating from this distribution by using an alternative PDF $p(z) \neq P^{B_c}(x, z)$ introduces a cumulative product of importance weights $\prod_i \frac{P^{B_c}(x_{i-1}, z_i)}{p_i(z_i)}$, which typically leads to excessive variance and numerical instability, as illustrated in Fig. 6. For screened Poisson equations, we employ the standard (non-screened) Green's function and Poisson kernel as the importance sampling PDFs. This choice yields manageable variance while avoiding the prohibitively high computational cost of evaluating the normalization constants required for sampling directly from the screened counterparts (e.g., via rejection sampling).

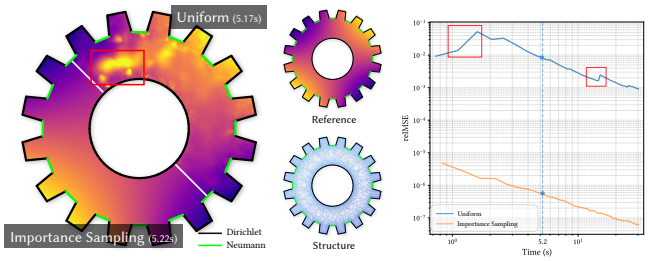


Fig. 6. Using uniform sampling instead of the normalized Poisson kernel $P^{B_c}(x, z)$ leads to excessive variance and significant artifacts (red boxes). In contrast, importance sampling (using the standard Poisson kernel) avoids weight accumulation, resulting in a smooth and rapid convergence of the solution.

Upon reaching the Dirichlet boundary, we traverse the path backward to recover the estimates of path points, updating the Fourier coefficients in Eq. (10).

4.5 Solution Reconstruction

Once the walking phase is complete, the solution at any query point is reconstructed. We identify all associated probes, i.e., $|x - c|/R_c < \alpha_{\text{rec}}$. For the Poisson equation, we sample $M = \max(\mu R_c^d, M_{\min})$ points in each probe to evaluate the contribution of the source term. The final solution is computed as the weighted average of the estimates provided by these probes, weighted by $w(x; B_c)$. Empirically, we set $M_{\min} = 1$ and select μ based on the scene scale.

4.6 Parallelization

Our implementation exploits parallelism to maximize performance. Probe Generation is executed serially as it accounts for a negligible fraction of the runtime. Random Walks are parallelized across threads; however, unlike traditional WoS, we employ thread-local caches during the reverse accumulation step (e.g., updating Fourier coefficients) to prevent race conditions. Finally, Reconstruction is naturally data-parallel, as each query point is processed independently without synchronization.

5 Experiments

For clarity, we denote the full algorithm described in Section 4 as **WoP**, and the approach of initiating probe-based walks from pixel centers and evaluating their starting points as **WoP walking** in the following text.

We evaluate the performance of WoP on various boundary value problems. We implement our method in C++ using FCPW [Sawhney 2021] for geometric queries. All experiments are conducted on a machine equipped with an Intel Core i9-13900HX CPU (24 cores, 32 threads) and 32 GB of RAM. For fair comparison, we evaluate the *initial results* of all methods (WoP, MVC [Bakbouk and Peers 2023], HC [Zhou et al. 2025], and WoSt [Miller et al. 2024b]) without secondary refinement passes, although our method is equally compatible with such post-processing.

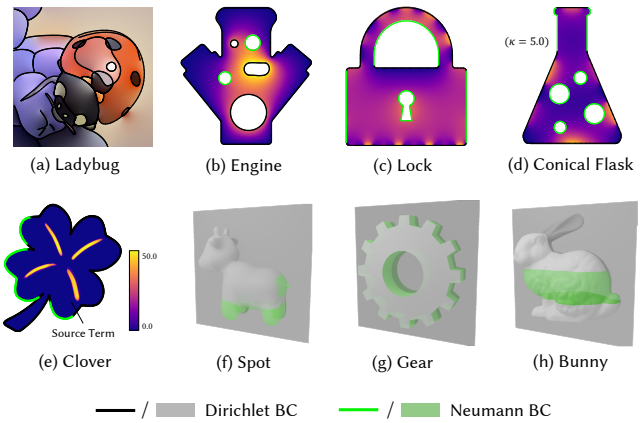


Fig. 7. Our experimental dataset with different boundary conditions. For 3D scenes (f-h), we adopt solving on slices as the visualization method.

5.1 Ablation Study

Effectiveness of Path Reuse. Table 1 analyzes variance reduction by comparing our full method against three variants: (i) *Only Reused*, which only uses intermediate path points to estimate the Fourier coefficients; (ii) *w/o Reused*, which only uses the initial starting points; (iii) *w/o Reused & WoP walking*, which replaces WoP walking with standard WoSt (similar to the HC method). The performance gain stems primarily from Path Reuse; enabling it (*Full* vs. *w/o Reused*) reduces error by an order of magnitude under equal computation time. Additionally, when excluding reused samples, replacing WoP walking with WoSt introduces no significant gain or degradation in terms of error. It is expected because of the numerical equivalence of WoP walking and WoSt.

Self-normalization and Control Variates. As shown in Table 2, combining self-normalization (SN) and control variates (CV) yields the lowest error across all scenes. Fig. 8 visualizes that SN+CV produces the smoothest solution with minimal error with negligible computational overhead in the Dirichlet-dominated scene *Puzzle*. Furthermore, CV remains effective even when integrated into HC.

Truncation Order L. Fig. 15 shows the relationship between relMSE and the truncation order L. The error converges to a fixed level beyond a certain L. Regarding performance, the *Accumulate* time is largely insensitive to L, whereas the *Query* time is proportional to it. Since $L = 10$ provides a consistent balance between accuracy and efficiency across all scenes, we use it for all our experiments.

Walking Ratio α_{walk}. As α_{walk} is typically smaller than the reconstruction ratio α_{rec}, it indirectly controls probe density. Given a fixed time budget, excessive probe overlap slows convergence. Fig. 16 shows that $\alpha_{\text{walk}} = 0.6$ achieves optimal efficiency.

WoSt Fallback probability p_{fb}. Random walks in WoP generally require more steps to reach the boundary compared to WoSt. To quantify this, we measure the path lengths of WoP walking and WoSt in Fig. 12. While longer paths provide more samples for reuse, they also increase the per-walk computational cost. Fig. 13 demonstrates

Table 1. Ablation study on the Lock (2D) and Gear (3D) scenes. We report the relMSE comparing four configurations: *Full* (all samples), *Only Reused* (reused path samples only), *w/o Reused* (initial samples only, excluding path reuse), and *w/o Reused and WoP walking* (initial samples only and replacing the WoP walking step with WoSt, similar to HC). Bold values indicate the best performance.

Scene	Lock (2D)				Gear (3D)			
	Equal Iterations		Equal Time		Equal Iterations		Equal Time	
	1 Iters	30 Iters	~ 4.80 s	~ 30.00 s	1 Iters	36 Iters	~ 21.00 s	~ 122.00 s
Full	2.603e-04	**1.219e-05**	8.198e-05	**1.623e-05**	**1.437e-03**	**7.869e-05**	**2.727e-04**	**7.045e-05**
Only Reused	**2.216e-04**	1.246e-05	**7.928e-05**	1.671e-05	3.656e-03	1.651e-04	6.014e-04	1.515e-04
w/o Reused	3.233e-03	1.196e-04	1.130e-03	2.356e-04	1.031e-02	3.389e-04	1.782e-03	3.111e-04
w/o Reused and WoP walking	3.138e-03	1.175e-04	1.146e-03	2.636e-04	1.038e-02	3.540e-04	2.040e-03	3.200e-04

Table 2. *Self-normalization* (SN) and *control variates* (CV) reduce variance consistently. The benefit of CV also extends to HC.

Scene	Puzzle		Engine	
	Equal Iter. 15 Iters	Equal Time ~ 10.0s	Equal Iter. 40 Iters	Equal Time ~ 10.0s
Full	**8.001e-07**	**1.785e-06**	**8.086e-06**	**3.658e-05**
w/o SN	5.248e-06	1.206e-05	1.142e-05	4.858e-05
w/o CV	8.458e-06	2.249e-05	1.063e-05	3.711e-05
w/o SN&CV	1.268e-05	3.224e-05	1.140e-05	6.541e-05
HC+CV	–	4.648e-06	–	1.825e-04
HC	–	6.726e-06	–	1.898e-04

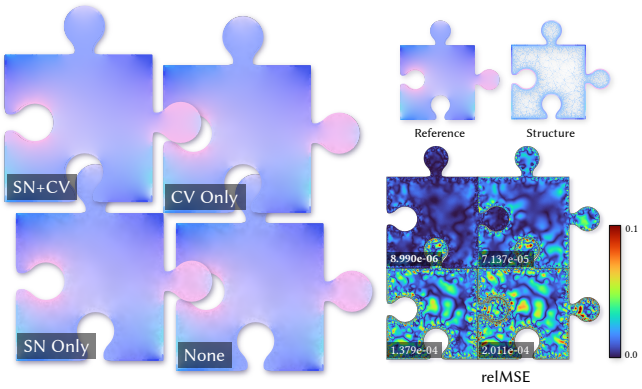


Fig. 8. Ablation study of *Self-Normalization* (SN) and *Control Variates* (CV) in the Dirichlet-dominated scene *Puzzle*. The combination of SN and CV achieves the smoothest reconstruction and the lowest relMSE. Since both SN and CV introduce negligible computational overhead, all results are obtained in 1 iteration (~2.30s) using a single thread.

that an appropriate fallback probability p_{fb} effectively balances this trade-off and accelerates convergence, though the optimal value remains scene-dependent.

5.2 Comparison with other Variance Reduction Methods

We evaluate WoP against HC and MVC on 2D and 3D Laplace, Poisson, and screened Laplace equations (visualized in Fig. 7). Figs. 1, 4 and 11 compares WoP, WoSt, MVC, and HC across various boundary conditions. By leveraging path information, WoP significantly reduces error in less time, achieving the best performance across all scenarios. Fig. 14 illustrates error convergence: WoP maintains a decay rate consistent with WoSt and HC. Despite higher initialization overhead for data structures, WoP delivers the lowest error from the first iteration.

We visualize the spatial distribution of reused samples in Fig. 9, where our method achieves a significantly higher density compared to HC and MVC. While baseline methods suffer from undersampling near Neumann boundaries—ignoring the slower convergence in these regions—our approach turns this challenge into an advantage. The longer random walks inherent to these areas naturally translate into more reused samples, yielding a more stable solution.

As shown in Fig. 10, WoP consumes more memory than alternatives to store path information. This cost scales with parallelization, as each thread must maintain its own path data.

Furthermore, to analyze the runtime bottlenecks in parallel execution, we provide a runtime breakdown in Figs. 17 and 18. The results indicate that the *Compute* pass accounts for the majority of total runtime. Within this pass, the overhead is dominated by the *Fallback WoSt* component. This is particularly evident in Neumann-dominated scenes, where fallback is triggered more frequently, and in 3D scenarios where probe coverage is localized near the ROI plane, resulting in a higher fallback ratio.

6 Conclusion, Limitations and Future Work

We have presented a high-performance variance reduction framework for the Walk on Spheres (WoS) algorithm by exploiting the Markov property of random walks to reuse path information. Our method seamlessly integrates a "Walk on Probes" (WoP) strategy with local series expansions, enabling both efficient path collection and accurate solution reconstruction. This approach is versatile, supporting (screened) Poisson equations with mixed Dirichlet and Neumann boundary conditions. Our results demonstrate that our method significantly enhances efficiency and robustness compared to state-of-the-art techniques, while fully retaining the mesh-free advantage of being immune to mesh quality issues.

Limitations. The primary limitation of WoP is its significant memory footprint. Unlike standard WoS, where contributions are accumulated on the fly, our method requires a deferred backward evaluation to reconstruct solutions at intermediate path points. This

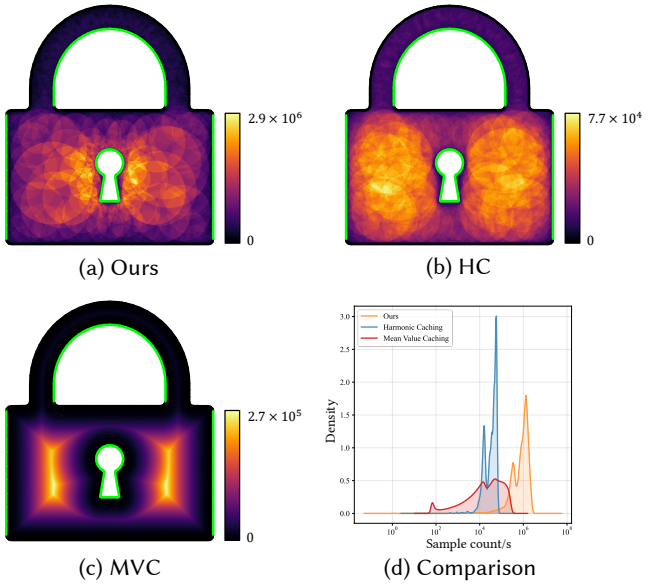


Fig. 9. Left: Reused sample count at each query point. Our method provides more samples, especially in the regions near the Neumann boundaries, where HC and MVC faces undersampling. Right: The distribution of reused samples per query point.

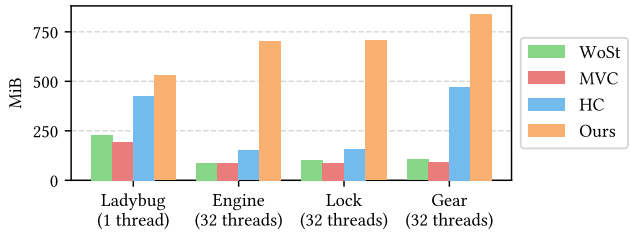


Fig. 10. Peak memory consumption (MiB) for different solvers over 10 iterations. Our method exhibits higher memory overhead, especially during parallel execution.

necessitates caching the full trajectory information—including probe indices, sampled angles, PDFs, and source/boundary contributions—until the walk terminates. Since a single walk may involve up to 10,000 steps, the per-thread storage requirement becomes substantial. This memory constraint currently hinders massive parallelism, posing challenges for efficient GPU implementation due to limited register and shared memory resources.

Future Work. Several promising directions remain. First, the intermediate estimates collected by WoP on pre-generated probes effectively provide a coarse solution representation; this information could be leveraged to enable path guiding strategies, similar to recent advances by Huang et al. [2025]. Second, generalizing the geometry from spherical probes to more adaptive shapes—such as star-shaped regions for better Neumann support, or semi-spheres and sectors for better boundary alignment [Himmler and Günther 2025]—could reduce the number of primitives and mitigate near-boundary geometric errors. Finally, applying our framework to variable-coefficient PDEs remains a non-trivial challenge, as it requires deriving corresponding local series expansions for spatially varying coefficients.

Acknowledgments

This work was supported by Laoshan Laboratory (No. LSKJ202300305) and the National Natural Science Foundation of China (U25A20444, 62502484, 62025207), the Provincial Natural Science Foundation of Anhui, PR China (2408085QF197), and the Fundamental Research Funds for the Central Universities (WK0010000095).

We thank all the reviewers for their constructive comments and suggestions. We also thank Tianyu Huang from Tsinghua University for many valuable discussions.

Finally, we thank the providers of the test scenes and models used in this work, including the *Ladybug* scene from Orzan et al. [2008], the *Engine* model from *Zombie* [Sawhney and Miller 2023], the *Bunny* model from the Stanford Computer Graphics Laboratory, the *Spot* model from Crane et al. [2013], and the *Watch Gear* scene [The Vipron 2024]. The rest of the test scenes were created by the authors using Blender [Blender Online Community 2018]; we are grateful to the Blender Foundation and the open-source community for providing such a powerful platform.

References

Ghada Bakbouk and Pieter Peers. 2023. Mean Value Caching for Walk on Spheres. In *Eurographics Symposium on Rendering*, Tobias Ritschel and Andrea Weidlich (Eds.). The Eurographics Association. doi:10.2312/sr.20231120

Anchang Bao, Jie Xu, Enya Shen, and Jianmin Wang. 2025. Off-Centered WoS-Type Solvers with Statistical Weighting. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, Article 127, 11 pages. doi:10.1145/3757377.3763852

Blender Online Community. 2018. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam. http://www.blender.org

Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2013. Robust fairing via conformal curvature flow. *ACM Trans. Graph.* 32, 4, Article 61 (July 2013), 10 pages. doi:10.1145/2461912.2461986

Michael Czekanski, Benjamin Faber, Margaret Fairborn, Adelle Wright, and David Bindel. 2024. Walking on Spheres and Talking to Neighbors: Variance Reduction for Laplace's Equation. *arXiv e-prints* (April 2024). arXiv:2404.17692 [physics.comp-ph] doi:10.48550/arXiv.2404.17692

B. S. Elepov and G. A. Mikhailov. 1969. Solution of the Dirichlet Problem for the Equation $\Delta u - cu = -q$ by a Model of "Walks on Spheres". *U. S. S. R. Comput. Math. and Math. Phys.* 9, 3 (1969), 194–204. doi:10.1016/0041-5553(69)90070-6

Lawrence C Evans. 2022. *Partial differential equations*. Vol. 19. American mathematical society.

G. Greger, P. Shirley, P.M. Hubbard, and D.P. Greenberg. 1998. The Irradiance Volume. *IEEE Computer Graphics and Applications* 18, 2 (March 1998), 32–43. doi:10.1109/38.656788

Paul Himmler and Tobias Günther. 2025. Conformal First Passage for Epsilon-free Walk-on-Spheres. *ACM Trans. Graph.* 44, 4, Article 40 (July 2025), 11 pages. doi:10.1145/3730942

Tianyu Huang, Jingwang Ling, Shuang Zhao, and Feng Xu. 2025. Guiding-Based Importance Sampling for Walk on Stars. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, Article 3, 12 pages. doi:10.1145/3721238.3730593

Mark Kac. 1949. On Distributions of Certain Wiener Functionals. *Trans. Amer. Math. Soc.* 65, 1 (1949), 1–13.

Shizuo Kakutani. 1944a. Two-Dimensional Brownian Motion and Harmonic Functions. *Proceedings of the Imperial Academy* 20, 10 (1944), 706 – 714. doi:10.3792/pia/1195572706

Shizuo Kakutani. 1944b. Two-dimensional Brownian Motion and Harmonic Functions. *Proceedings of the Imperial Academy* 20, 10 (1944), 706–714. doi:10.3792/pia/1195572706

J. Krivanek, P. Gautron, S. Pattanaik, and K. Bouatouch. 2005. Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics* 11, 5 (2005), 550–561. doi:10.1109/TVCG.2005.83

Zander Majercik, Jean-Philippe Guertin, Derek Nowrouzezahrai, and Morgan McGuire. 2019. Dynamic Diffuse Global Illumination with Ray-Traced Irradiance Fields. *Journal of Computer Graphics Techniques (JCGT)* 8, 2 (5 June 2019), 1–30. http://jcgt.org/published/0008/02/01/

Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2023. Boundary Value Caching for Walk on Spheres. *ACM Trans. Graph.* 42, 4, Article 82 (July 2023), 11 pages. doi:10.1145/3592400

Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2024a. Differential Walk on Spheres. *ACM Trans. Graph.* 43, 6, Article 174 (Nov. 2024), 18 pages. doi:10.1145/3687913

Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2024b. Walkin' Robin: Walk on Stars with Robin Boundary Conditions. *ACM Trans. Graph.* 43, 4, Article 41 (July 2024), 18 pages. doi:10.1145/3658153

Mervin E. Muller. 1956. Some Continuous Monte Carlo Methods for the Dirichlet Problem. *The Annals of Mathematical Statistics* 27, 3 (1956), 569–589. http://www.jstor.org/stable/2237369

Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. 2008. Diffusion Curves: A Vector Representation for Smooth-Shaded Images. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 1–8. doi:10.1145/1360612.1360691

Yang Qi, Dario Seyb, Benedikt Bitterli, and Wojciech Jarosz. 2022. A Bidirectional Formulation for Walk on Spheres. *Computer Graphics Forum* 41, 4 (2022), 51–62. doi:10.1111/cgf.14586

Karl Sabelfeld, Alexander I Levykin, and Irina Shalimova. 2008. Random Walk on Fixed Spheres method for electro-and elastostatics problems. (2008).

Karl Sabelfeld, Irina Shalimova, and Alexander Levykin. 2006. Random walk on fixed spheres for Laplace and Lamé equations. (2006).

Karl K Sabelfeld, Irina A Shalimova, and Alexander I Levykin. 2004. Discrete random walk on large spherical grids generated by spherical means for PDEs. *Monte Carlo Methods & Applications* 10 (2004).

Rohan Sawhney. 2021. *FCPW: Fastest Closest Points in the West.*

Rohan Sawhney and Keenan Crane. 2020. Monte Carlo Geometry Processing: A Grid-Free Approach to PDE-based Methods on Volumetric Domains. *ACM Trans. Graph.* 39, 4, Article 123 (Aug. 2020), 18 pages. doi:10.1145/3386569.3392374

Rohan Sawhney and Bailey Miller. 2023. *Zombie: Grid-Free Monte Carlo Solvers for Partial Differential Equations.*

Rohan Sawhney, Bailey Miller, Ioannis Gkioulekas, and Keenan Crane. 2023. Walk on Stars: A Grid-Free Monte Carlo Method for PDEs with Neumann Boundary Conditions. *ACM Trans. Graph.* 42, 4, Article 80 (July 2023), 20 pages. doi:10.1145/3592398

Rohan Sawhney, Dario Seyb, Wojciech Jarosz, and Keenan Crane. 2022. Grid-Free Monte Carlo for PDEs with Spatially Varying Coefficients. *ACM Trans. Graph.* 41, 4, Article 53 (July 2022), 17 pages. doi:10.1145/3528223.3530134

Peter-Pike Sloan, Jan Kautz, and John Snyder. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.* 21, 3 (July 2002), 527–536. doi:10.1145/566654.566612

Ryusuke Sugimoto, Terry Chen, Yiti Jiang, Christopher Batty, and Toshiya Hachisuka. 2023. A Practical Walk-on-Boundary Method for Boundary Value Problems. *ACM Trans. Graph.* 42, 4, Article 81 (July 2023), 16 pages. doi:10.1145/3592109

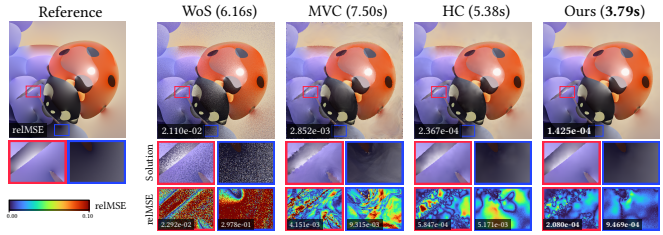
The Vipron. 2024. Mechanical Watch Mechanism. https://sketchfab.com/3d-models/mechanical-watch-mechanism-56e60ad7abf94f16a9dbf92f319ffdf7. Accessed: Jan 22, 2026. Available under CC BY 4.0..

Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. 1988. A ray tracing solution for diffuse interreflection. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '88)*. Association for Computing Machinery, New York, NY, USA, 85–92. doi:10.1145/54852.378490

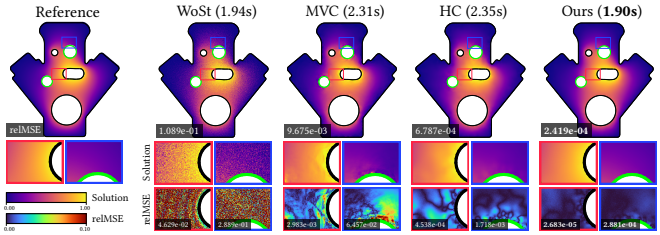
Ekrem Fatih Yilmazer, Delio Vicini, and Wenzel Jakob. 2024. Solving Inverse PDE Problems using Grid-Free Monte Carlo Estimators. *ACM Trans. Graph.* 43, 6, Article 175 (Nov. 2024), 18 pages. doi:10.1145/3687990

Zihan Yu, Lifan Wu, Zhiqian Zhou, and Shuang Zhao. 2024. A Differential Monte Carlo Solver For the Poisson Equation. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) *(SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, Article 11, 10 pages. doi:10.1145/3641519.3657460

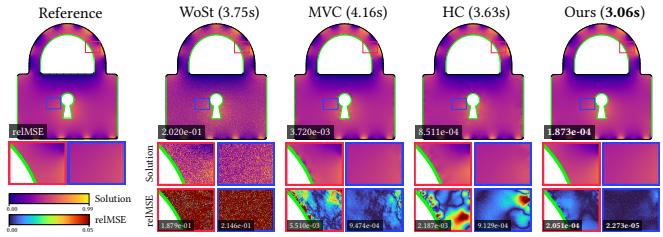
Zihong Zhou, Eugene d'Eon, Rohan Sawhney, and Wojciech Jarosz. 2025. Harmonic Caching for Walk on Spheres. *ACM Trans. Graph.* 44, 6, Article 253 (Dec. 2025), 15 pages. doi:10.1145/3763322



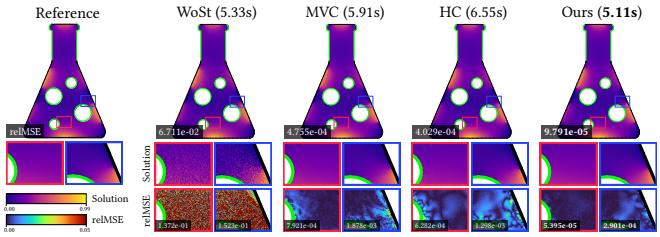
(a) *Ladybug*: pure Dirichlet BC, running with single thread.



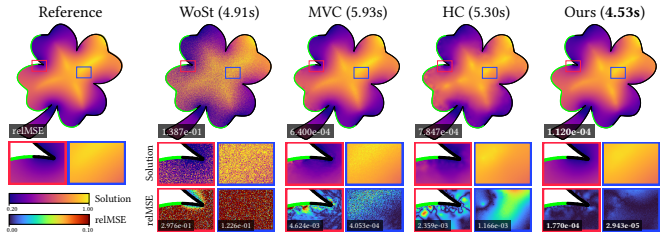
(b) *Engine*: mixed Dirichlet and Neumann BC.



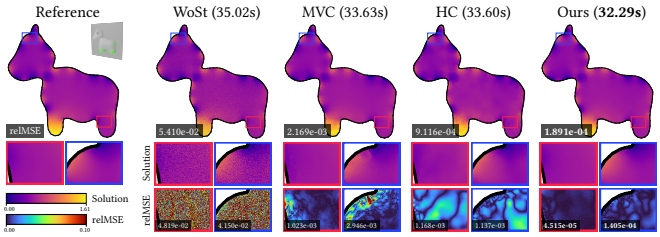
(c) *Lock*: mixed Dirichlet and Neumann BC with more Neumann Boundaries.



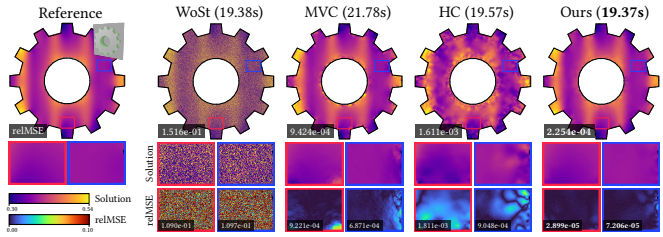
(d) *Conical Flask*: mixed Dirichlet and Neumann BC with screening parameter $\kappa = 5.0$.



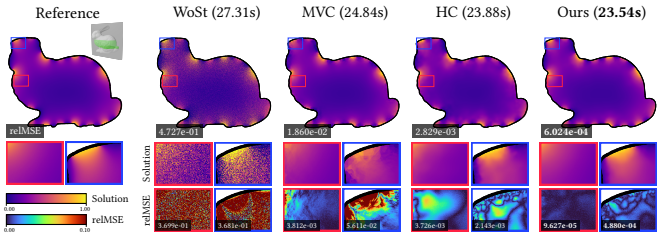
(e) *Clover*: mixed Dirichlet and Neumann BC with source term.



(f) *Spot*: 3D scene with mixed Dirichlet and Neumann BC.



(g) *Gear*: 3D scene with mixed Dirichlet and Neumann BC, and more Neumann Boundaries.



(h) *Bunny*: 3D scene with mixed Dirichlet and Neumann BC with screening parameter $\kappa = 5.0$.

Fig. 11. Comparison of variance reduction techniques across various 2D and 3D boundary value problems. For 3D scenes, we set query points only on the interested planar. For MVC, we follow the adaption described in [Zhou et al. 2025, Appendix E] to handle screened Laplace equations.

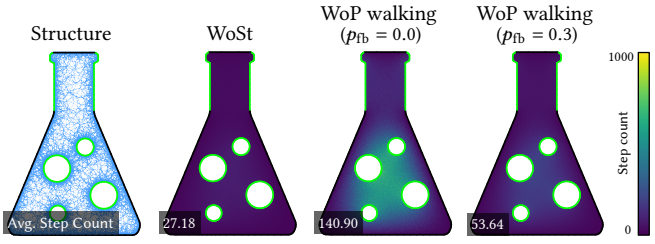


Fig. 12. In Neumann-dominated scenes (e.g., *Conical Flask*), WoP walking generates longer walking paths than WoSt, yielding more samples for reuse but also increasing computational cost. Increasing the fallback probability p_{fb} shortens the paths significantly, achieving faster convergence.

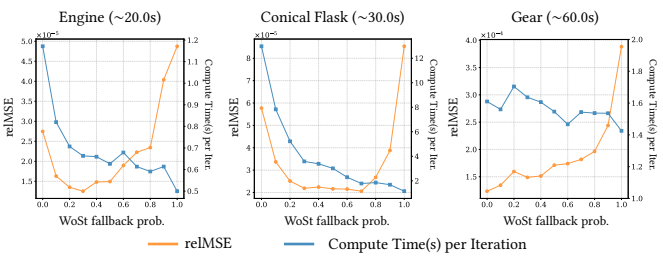


Fig. 13. Higher *WoSt fallback probability* reduces reused samples, but also shortens the random walking path and saves compute time per iteration. The optimal p_{fb} varies across different scenes.

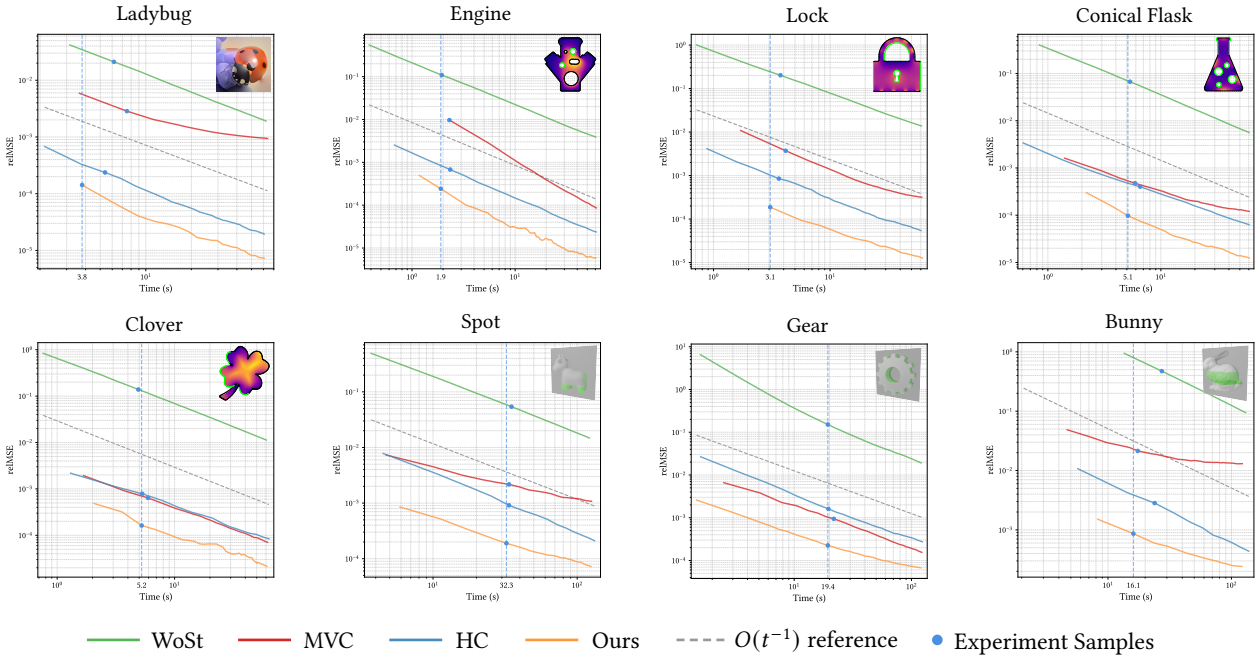


Fig. 14. relMSE vs. time in log-log scale. Our method exhibits faster and more stable convergence across all tested scenes compared to state-of-the-art baselines.

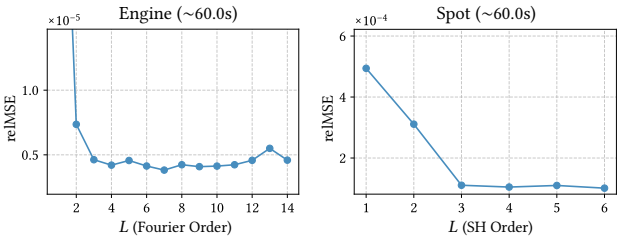


Fig. 15. Influence of truncation order on convergence. The relMSE is plotted against the Fourier order L for 2D (Engine) and Spherical Harmonics (SH) order L for 3D (Spot) scenes. Low truncation order leads to premature convergence at a higher error floor due to truncation bias.

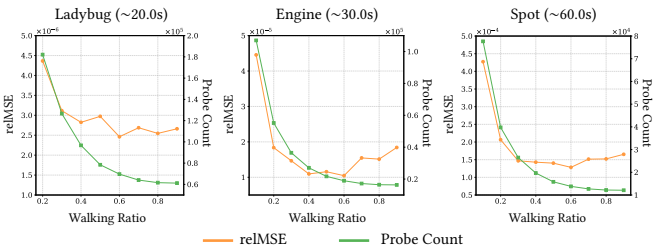


Fig. 16. Lower *walking ratio* yields denser probes and slower convergence. $\alpha_{\text{walk}} = 0.6$ yields the best performance.

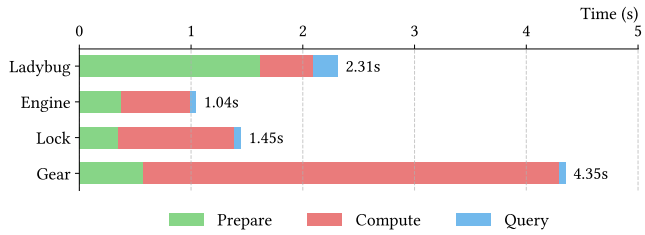


Fig. 17. Runtime breakdown of the three passes in our solver. Each scene is evaluated using 32 threads over 10 iterations; the reported *Compute* time represents the average across all iterations.

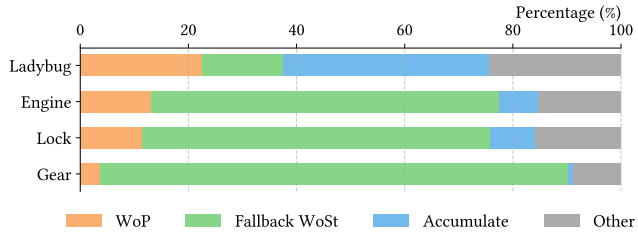


Fig. 18. Detailed breakdown of the *Compute* pass across different scenes. The execution time is categorized into: (i) *WoP*, involving probe-related geometric queries and sampling; (ii) *Fallback WoSt*, involving geometric queries and sampling for WoSt steps; and (iii) *Accumulate*, involving back-propagation and Fourier series expansion.